Smartphones: Privacy Standpoint

Jagdish Prasad Achara, Claude Castelluccia, James-Douglass Lefruit, Vincent Roca

Smartphones: Privacy Standpoint

as part of

Mobilitics (Inria-CNIL project)

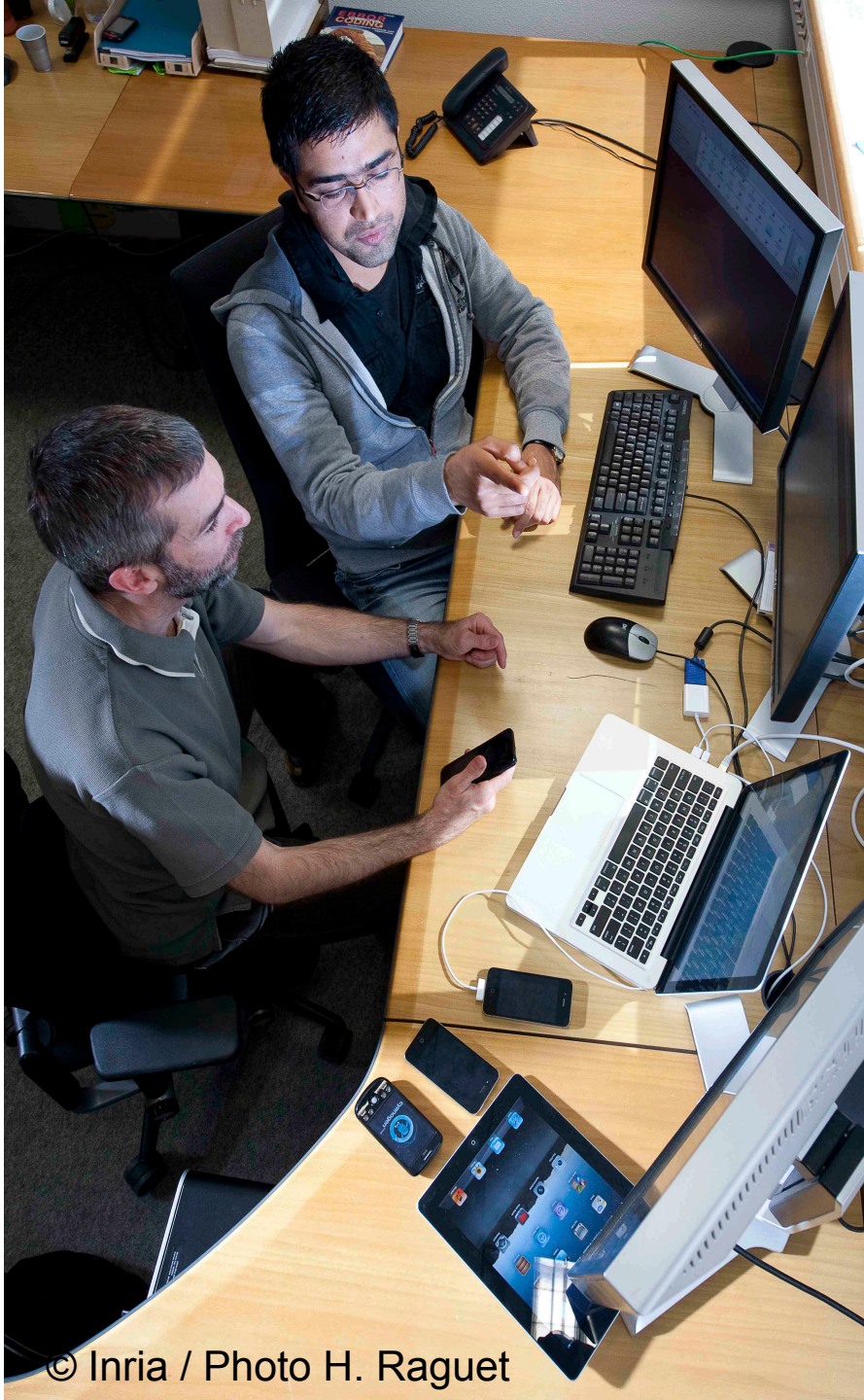
Jagdish Prasad Achara[Speaker],
Claude Castelluccia, James-Douglass Lefruit, Vincent Roca
(**Privatics team,** Inria Rhone-Alpes)

Workshop on Location-Related Services, EIT ICT Labs,
Saarbrücken, Dec 5th, 2013

informatiques mathématiques

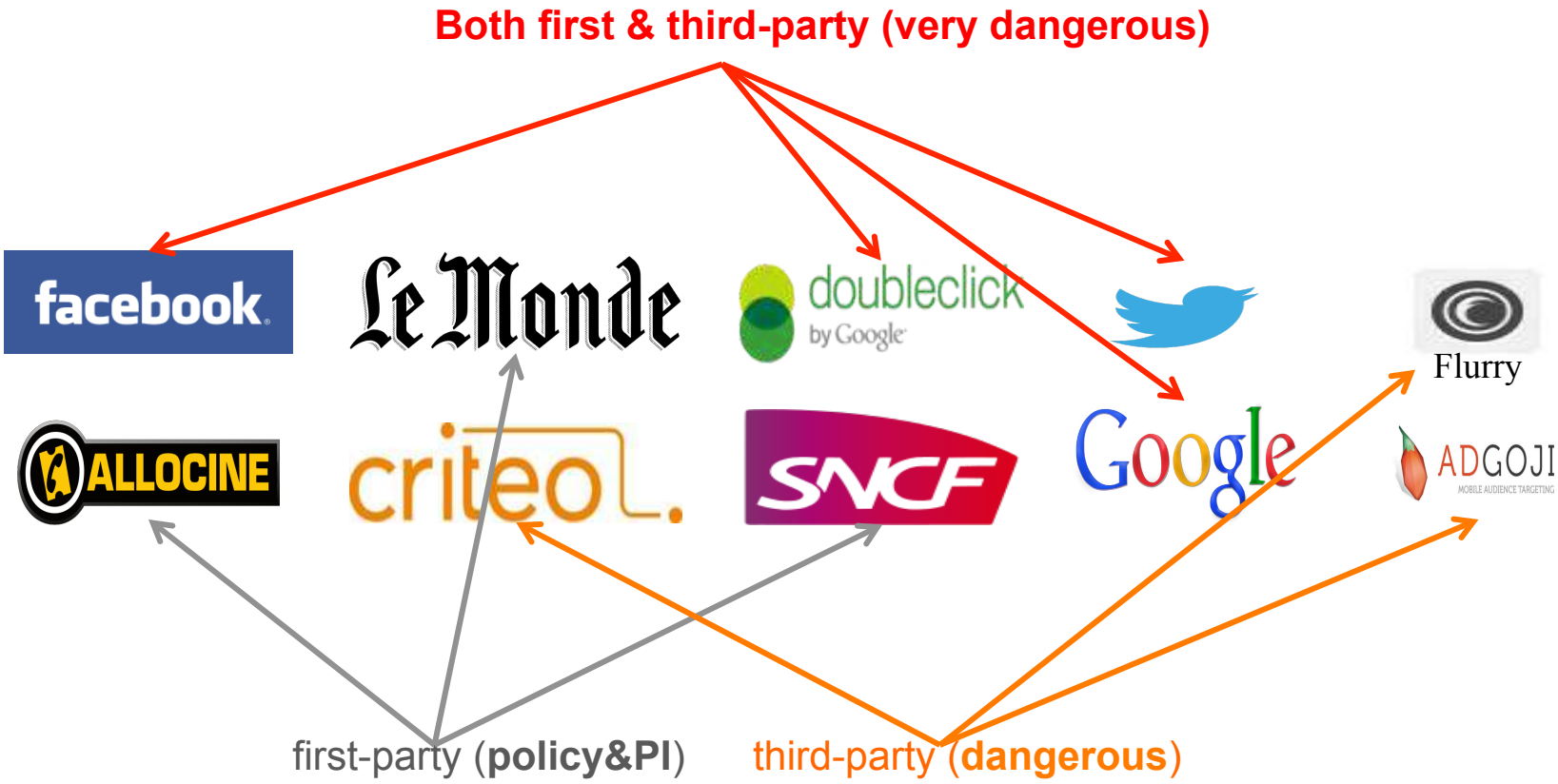
Outline of the talk

- ***Motivations & Goals***
- Our “Tracking the trackers” methodology
- Results obtained
- Conclusions and Future work



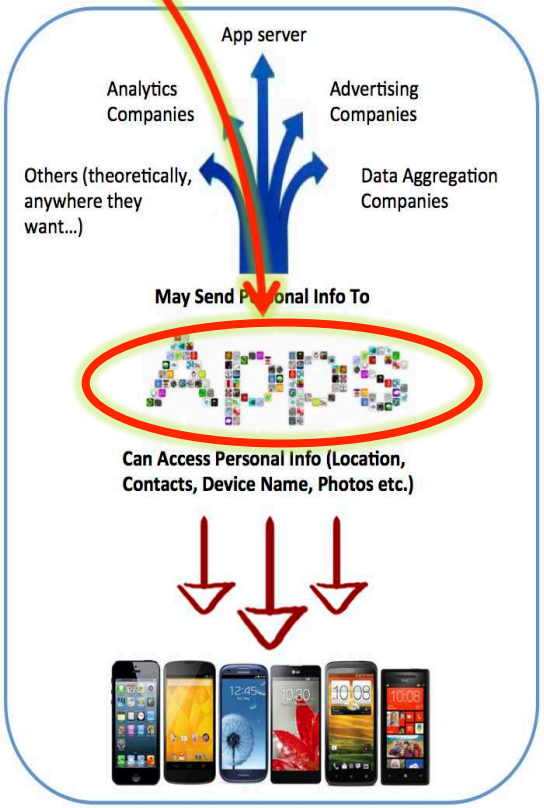
Presence of a large number of actors

- Due to revolutionary arrival of AppStore model
  - No more the presence of merely GSM/CDMA provider
  - Both first-party (App provider) & third-party(Advertisers, Analytics companies etc.)



More possibilities for PI leakage to various parties

- **Not only limited to web browsers** as is the case in desktops/laptops
  - Apps for dedicated services (FB, LeMonde, SNCF etc.)



Difficult to trust all these parties

- various scandals in the past
  - For example, Twitter and Path uploading users all contacts to their servers [1] [2]
- WSJ: What they know – Mobile [3]

[1] http://mclov.in/2012/02/08/path-uploads-your-entire-address-book-to-their-servers.html

[2] http://www.theregister.co.uk/2012/02/15/twitter_stores_address_books/

[3] http://blogs.wsj.com/wtk-mobile/

Smartphones are well suited to marketers/ trackers

- contain a lot of info on user **interests** and **behaviors** because
  - various sensors (GPS, camera etc) and comm technologies (WiFi, GSM etc.) generate PI
  - smartphones are at the **center of our cyber activities** and **very personal** (not shared)
  - smartphones have almost **all-time Internet connectivity**
  - they're barely turned off

→ leads to accurate and detailed user profiling

A direct consequence is a large presence of online advertisers/trackers

Flurry

and many others…

→→→This necessitates scrutinizing smartphones for privacy risks

"tracking the trackers"

Mobilitics project and its goals

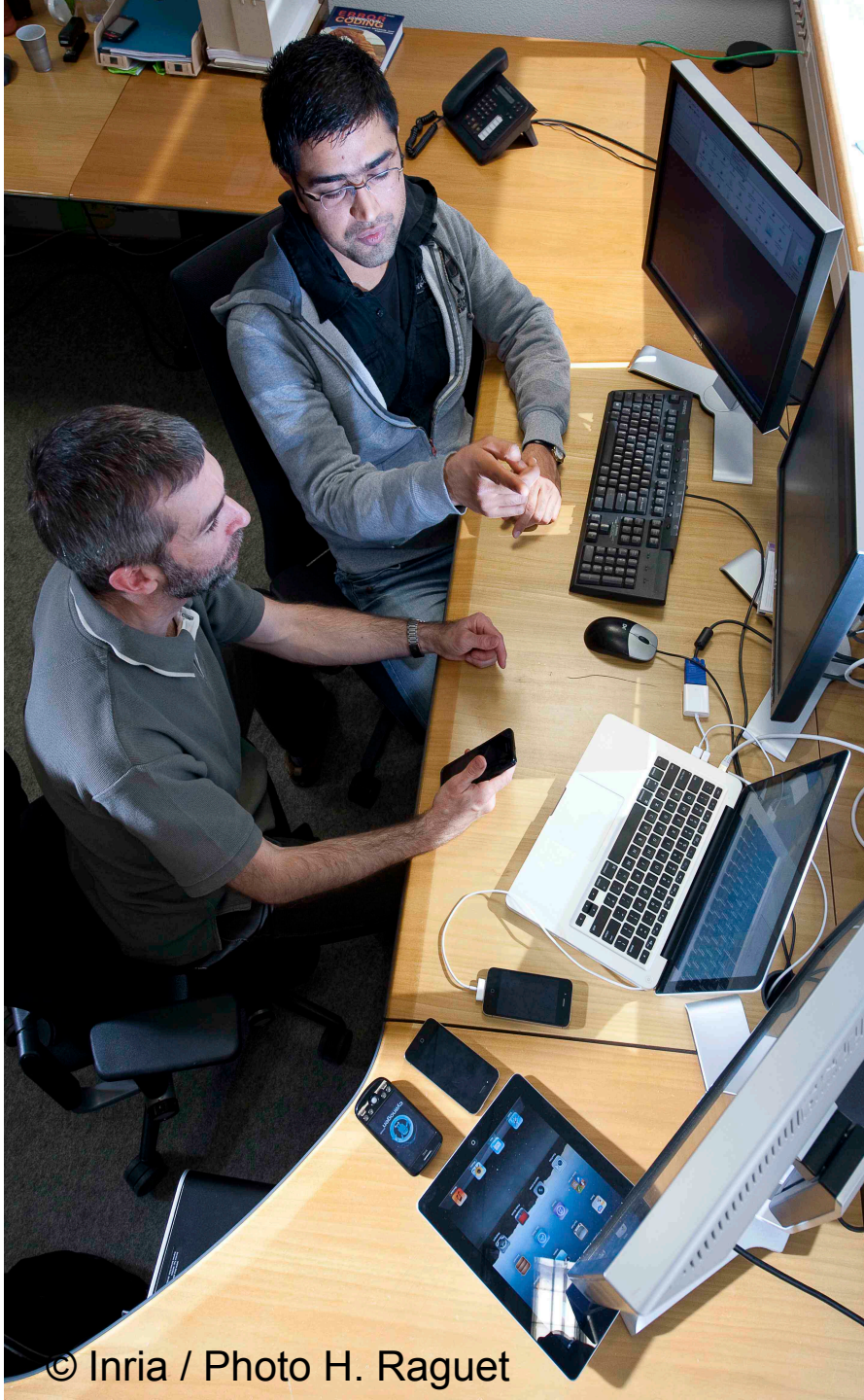
- started in January 2012

- focuses on Android and iOS
  - the leading Smartphone OS

- **Goals**: investigate smartphone Apps and OS for potential privacy risks…

Outline of the talk

- *Motivations & Goals*
- **Our “tracking the trackers” methodology**
- Results obtained
- Conclusions and Future work



General approach (iOS & Android)

1. Run Apps on instrumented versions of Android and iOS

2. Collect and store data related to the access to user PI along with inputs to data modification APIs and all the network traffic (plain-text or SSL) in a local SQLite database

3. Post-analysis of data collected

iOS (1): Some background

- Enforcement of user privacy by Apple in two steps
  1. Apple vetting process when Apps are submitted to AppStore
  2. Users are asked before iOS gives access to user PI to an App

- Closed source and only code signed from Apple can be executed
  - enforced by secure boot chain

- Also, no App source available → only binary rewriting is possible

iOS (2): Some background

- iOS Apps are written in
  - Objective-C, C, C++
- User PI can only be accessed through Apple's frameworks written in Objective-C/C/C++
  - … even if there are some exceptions (e.g. sysctl)
- Instrumenting iOS requires "**Jailbreaking**"
  - essentially a way to bypass Apple's secure boot chain

iOS (3): Realizing our general approach…

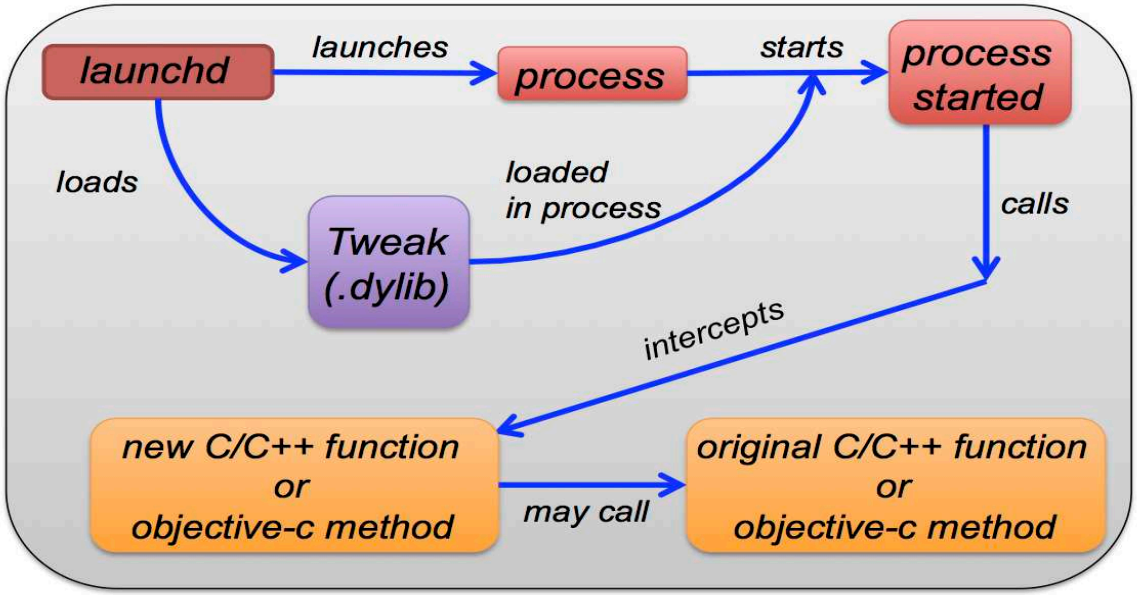
- As source code is not available, binary patching?
  - It's a nightmare, I think!

- Dynamically, at runtime?
  - Fortunately, yes!
    - **Use Objective-C runtime method "method_setImplementation"**
    - **Replace the C/C++ functions at assembly level.**

***NB: we use a third-party framework (MobileSubstrate) which makes it lot simpler… http://iphonedevwiki.net/index.php/MobileSubstrate**

iOS (4): Realizing our general approach…

- Whole code (modified implementation of the methods) is compiled in a dylib
  - and loaded at launch time in a process of interest



- We capture relevant info (method args, return values) and store it in a local SQlite DB

Android (1): Some background

- Apps are written mostly in Java but C/C++ can also be used with the help of JNI
- Java code is compiled to byte-code and then, converted from JVM-compatible .class files to Dalvik compatible .dex files
- These .dex files are executed in Dalvik Virtual Machine

Android (2): Realizing our general approach

- Change the source code itself
- Our custom code is added to APIs of interest to store the relevant data in a local SQLite DB
- We changed Android 4.1.1_r6 source code in our study

Post-Analysis of data (iOS & Android)

1. Identify **private data accessed** by Apps

2. Search for private data in the **network traffic** to see if it's sent, and where

3. Search for private data in the input to **cryptographic / hash functions**, and if there's some, search the output in the **network traffic**

4. **iOS Specific**: find out if Apps use **cross-App tracking** techniques by using the "UIPasteBoard" class

Limitations of our Approach

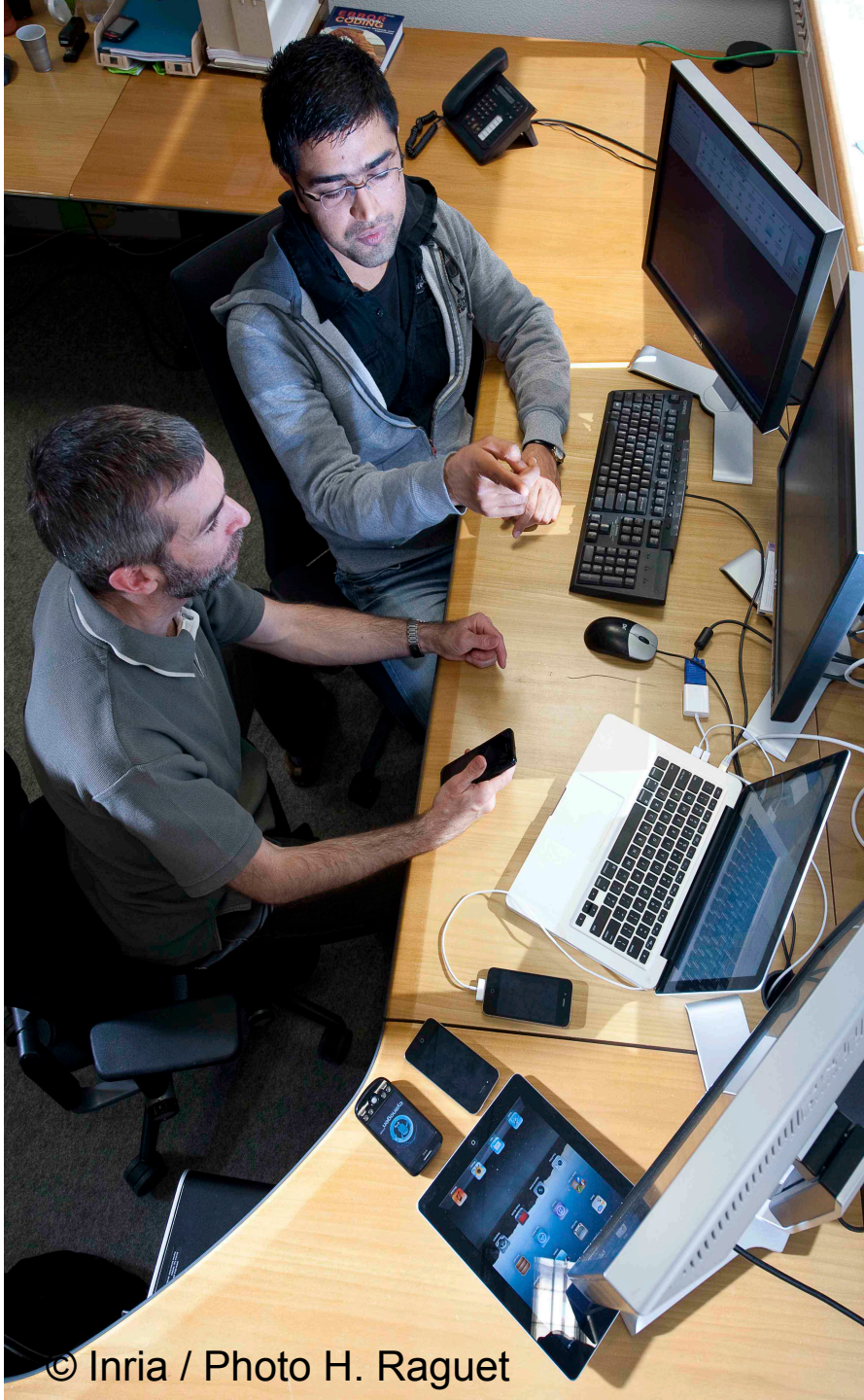
- Are private data manipulations (hash, encryption etc.) done with custom functions…
  - …rather than using standard iOS API?
  - if yes, we cannot detect it as we don't know what to search in the network traffic ☹
    - **e.g., a simple XOR with a static key is sufficient**

→ a **fundamental limitation** of our approach

- hard to evaluate if this is current practice or not
- But this means…results obtained using our technique would be **lower-bound**

Outline of the talk

- *Motivations & Goals*
- Our “tracking the trackers” methodology
- **Results obtained**
- Conclusions and Future work



Some facts before presenting Results

- We tested 140 free Apps available on both Android and iOS using our « tracking the trackers » methodology

- Experiments were carried out on iOS 6.1.2 and Android 4.1.1

- In our study, we consider user PI
  1. **Stable Identifiers**: that can be uniquely attached to users for tracking purposes
  2. **Any info revealing users' interests and behavior**

A glimpse of collection of unique identifiers by various parties: iOS

Server/Comm. type	AdIdentifier	UDID	DeviceName	WiFiMACAddress	WiFiMACAddressModified
facebook.com(SSL)	Yes				
testflightapp.com(SSL)	Yes				
amazonaws.com(plain-text)	Yes	Yes		Yes	
adjust.io(SSL)	Yes				
gameloft.com(plain-text)	Yes			Yes	
gameloft.com(SSL)	Yes			Yes	
amazonaws.com(SSL)	Yes			Yes	
paypal.com(SSL)	Yes		Yes		
boxcar.io(SSL)			Yes		
flurry.com(SSL)					Yes
tapjoy.com(SSL)	Yes				
jumptap.com	Yes				
mobile-adbox.com(SSL)	Yes				
fiksu.com(SSL)	Yes				
tapad.com(SSL)	Yes				
tapjoyads.com(SSL)	Yes			Yes	
tapjoyads.com(plain-text)		Yes			
appads.com(plain-text)	Yes				
adcolony.com(plain-text)	Yes				
sophiacom.fr(plain-text)	Yes				
smartadserver.com(plain-text)	Yes				
mopub.com(plain-text)	Yes				
swelen.com(plain-text)	Yes				
adtilt.com(plain-text)	Yes				
adtilt.com(SSL)	Yes				
booking.com(SSL)	Yes				
trademob.net(SSL)	Yes			Yes	
nanigans.com(plain-text)	Yes				
nanigans.com(SSL)	Yes				
ad-x.co.uk(SSL)	Yes				
eamobile.com(SSL)	Yes				
igstudios.in(plain-text)				Yes	
crittercism.com(SSL)			Yes		

A glimpse of collection of unique identifiers by various parties: Android

Server/Comm. type	Android ID	Phone No	IMEI	Serial No	IMSI	IMEI Modified	AndroidID Modified
engine.mobileapptracking.com(SSL)	Yes						
74.217.75.7(plain-text)	Yes						
iphone-mobilesite.airfrance.com(SSL)	Yes						
www.klm.com(SSL)	Yes						
mdotm.com(plain-text)	Yes		Yes				
mobage.com(plain-text)	Yes		Yes				
kochava.com(plain-text)	Yes		Yes				
msh.amazon.com(SSL)	Yes			Yes			
72.21.194.112(plain-text)	Yes			Yes			
google.com(SSL)	Yes						
smartadserver.com(plain-text)	Yes						
xiti.com(plain-text)	Yes						
badoo.com(SSL)	Yes		Yes		Yes		
ws.tapjoyads.com(SSL)	Yes		Yes			Yes	
playhaven.com(plain-text)	Yes						
adtilt.com(plain-text)	Yes		Yes				
yoz.io(plain-text)	Yes						
airbnb.com(SSL)	Yes						
groupon.com(SSL)	Yes						
fiksu.com(SSL)	Yes		Yes				
crittercism.com(SSL)	Yes		Yes				
googleapis.com(SSL)		Yes			Yes		
sstats.adobe.com(SSL)					Yes		
linode.com(plain-text)						Yes	
93.184.219.20(plain-text)							Yes
107.6.111.137(plain-text)							Yes
seattleclouds.com(plain-text)	Yes						
startappexchange.com(plain-text)	Yes						
91.103.140.6(plain-text)	Yes						
appwiz.com(SSL)	Yes	Yes	Yes				
airpush.com(SSL)	Yes	Yes				Yes	Yes
69.28.52.39(plain-text)	Yes		Yes				
209.177.95.171(plain-text)	Yes						
ad-market.mobi(plain-text)	Yes						
fastly.net(SSL)	Yes						

User PI collection: iOS

- **Again, on iOS, different kinds of user PI is sent to both first and third-parties** (out of a total of 140 free iOS Apps tested)

SIM Network name	Location	DeviceName	AddressBook	Accounts	SIM Number
testflightapp.com(SSL), clara.net(plain-text), capptain.com(SSL), groupon.de(plain-text), groupon.de(SSL), groupon.com(SSL), ebay.com(SSL), ec2-54-244-3-130.us-west-2.compute.amazonaws.com(plain-text)	bkt.mobi(plain-text), foursquare.com(SSL), groupon.de(SSL), voyages-sncf.com(plain-text), capptain.com(SSL)	paypal.com(SSL), crittercism.com(SSL), boxcar.io(SSL)	mobilevoip.com(plain-text)	twitter.com(SSL)	fring.com(SSL)

User PI collection: Android

- **Different kinds of user PI are sent to both first and third-parties** (out of a total of 140 free Android Apps tested)

Contacts	Location	Network Code	Operator Name	SIM Network code	WiFi AP Scan Info	Account Names
google.com(SSL)	seventynine.mobi(plain-text), platform.chekmein.com(SSL), airpush.com(SSL), appwiz.com(SSL), google.com(SSL), google.com(plain-text), 3g.cn(plain-text)	google.com(SSL), badoo.com(SSL), doubleclick.net(plain-text), appwiz.com(SSL), goforandroid.com(plain-text)	seventynine.mobi(plain-text), crittercism.com(SSL), msh.amazon.com(SSL), kiip.me(plain-text), 72.21.194.112(plain-text), badoo.com(SSL), ws.tapjoyads.com(SSL), adtilt.com(plain-text), groupon.com(SSL), groupon.de(SSL), 2o7.net(plain-text), m6replay.fr(SSL), appsflyer.com(SSL), airpush.com(SSL)	google.com(SSL), badoo.com(SSL), admarket.mobi(plain-text), goforandroid.com(plain-text), startappexchange.com(plain-text), appwiz.com(SSL)	badoo.com(SSL)	google.com(SSL), airpush.com(SSL), googleapis.com(SSL)

Leakage of App usage info: iOS

- **Various third-parties know what Apps a particular user is using**
  - It's like browsing history in case of web browsing

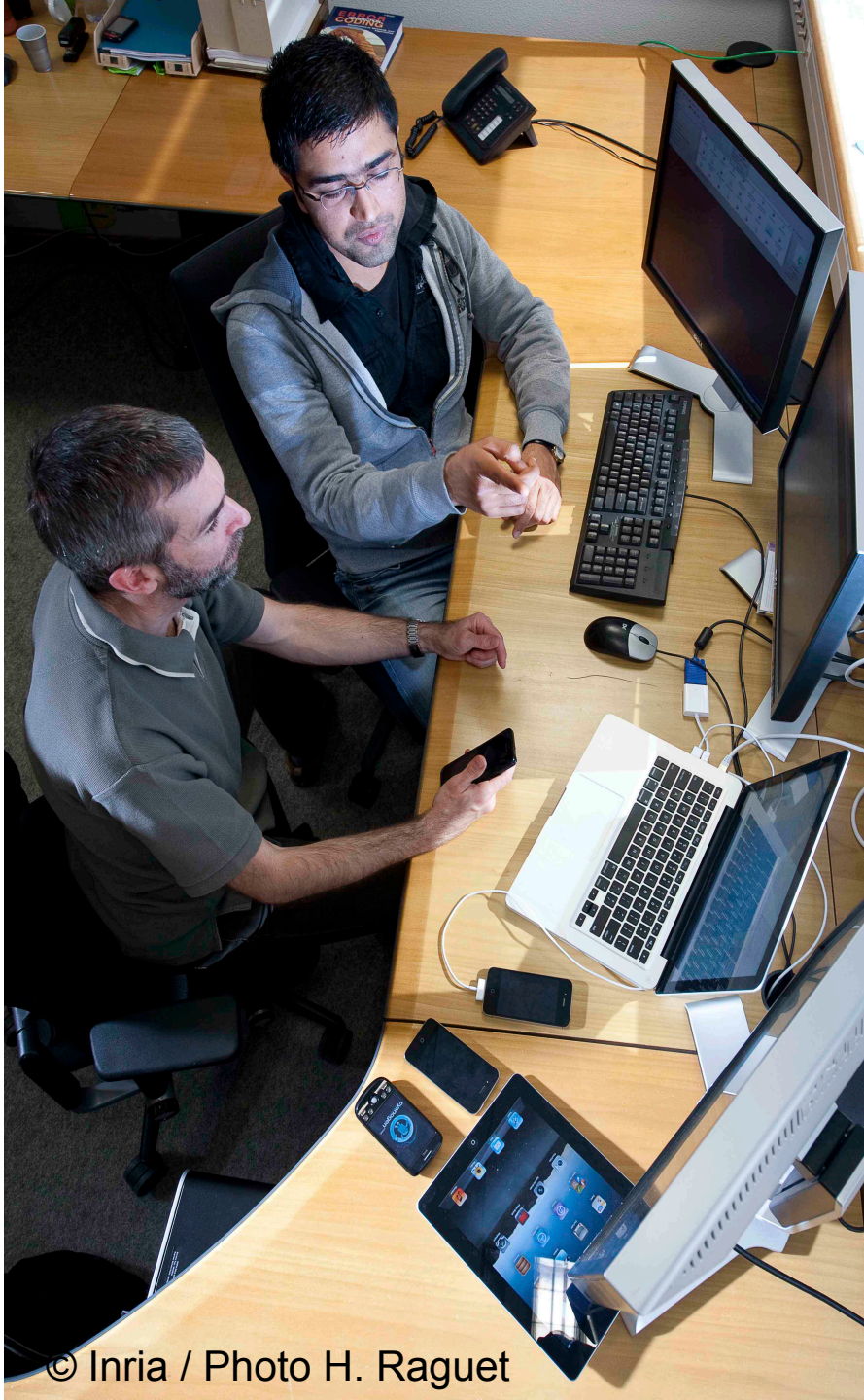
Third-party with type of Comm	Process Names
google-analytics.com(SSL)	InstantBeautyProduction, Evernote, LILIGO, Transilien, Viadeo, VDM, comuto, easyjet, VintedFR, Volkswagen
crashlytics.com(SSL)	dailymotion, TopEleven, AmazonFR, Path, RunKeeper, foodspotting, babbelSpanish, Deezer
urbanairship.com(SSL)	Wimbledon, RATP, HootSuite, DuplexA86, Appygraph, foodspotting, Volkswagen
flurry.com(plain-text)	TopEleven, Bible, RATP, Transilien, TripIt, DespicableMe, FlyAirIndia, Viadeo, Bankin', VDM, OCB, DuplexA86, SleepBot, Snapchat, Appygraph, Booking.com, foodspotting, Badoo, EDF-Releve, WorldCup2011, Quora, UrbanDictionary, babbelSpanish, MyLittleParis, Volkswagen
tapjoy.com(SSL)	TopEleven, Bible, DespicableMe, OCB, MCT
capptain.com(plain-text)	Viadeo, myTF1, rtl-fr-radios, 20minv3, iDTGV
xiti.com(plain-text)	laposte, ARTE, myTF1, lequipe, SoundCloud, 20minv3, Leboncoin
admob.com(plain-text)	VSC, BBCNews, WorldCup2011, RF12, UrbanDictionary

Leakage of App usage info: Android

Third-party with type of Comm	Process Names
google-analytics.com(SSL)	com.anydo, com.rechild.advancedtaskkiller, com.spotify.mobile.android.ui, com.google.android.googlequicksearchbox, com.dailymotion.dailymotion, com.aa.android, com.comuto, com.airbnb.android
doubleclick.net(plain-text)	com.tagdroid.android, com.rechild.advancedtaskkiller, bbc.mobile.news.ww, ua.in.android_wallpapers.spring_nature
trademob.com(SSL), google.com(SSL)	All the processes running on the phone
crashlytics.com(SSL)	com.evernote, com.path, com.lslk.sleepbot, com.twitter.android, com.dailymotion.dailymotion

Outline of the talk

- *Motivations & Goals*
- Our “tracking the trackers” methodology
- Results obtained
- **Conclusions and Future work**



Conclusions

- Private data is sent to various parties
  - As shown in the Tables before
- There is a clear need of better regulations to stop this practice.
- **A real problem today**
  - A user giving access to its PI to a particular App doesn't necessarily imply that he is ready to share it with other third-parties!

Future work

- We need to increase the number of Apps being tested to have a better idea of the phenomenon

- We must test **paid Apps too** to verify if some difference exist W.R.T. free Apps
  - How do free and paid versions of the same App differ from each other?

Questions/Remarks?

Thanks